BMTagger: Best Match Tagger for removing human reads from metagenomics datasets

Kirill Rotmistrovsky, Richa Agarwala*
NCBI/NLM, National Institutes of Health, Bethesda, MD 20894, USA
∗ E-mail: richa@helix.nih.gov

Abstract

The Human Microbiome Project (HMP) is sequencing genomes collected from various human body sites. The sequencing is being done using Illumina and 454 next-generation sequencing platforms that generate short reads in large volumes. As the sequences submitted contain reads for the human from whom the sample was collected, it is ethically necessary to provide the full set of sequences under controlled access portion of the Sequence Read Archive (SRA), and to provide only the sequences that have been screened for human "contamination" publicly. Best Match Tagger (BMTagger) is an efficient tool that discriminates between human reads and microbial reads without doing an alignment of all reads to the human genome.

In the sets we analyzed, BMTagger has less than 2% of the human reads classified as foreign and negligible (less than 0.01%) microbial reads classified as human. Aligning all reads to the human genome build 37 using BWA short read aligner shows that speed of BMTagger is comparable to that of BWA and that BWA does not report alignments for a large number of human reads even when there are only a few errors in the reads. The speed of BMTagger is partially explained by the observation that for most sets, only a small percentage of the reads need a determination by alignment. In our experiments with metagenomic datasets, the throughput of BMTagger ranges between 10 to 45 million paired 100 bp Illumina reads per hour on a single processor, and it is even better for 454 as longer reads are easier to discriminate. Using BMTagger facilitated a minimal delay between when the reads are submitted and when the screened subset becomes publicly available.

BMTagger is available at
ftp://ftp.ncbi.nlm.nih.gov/pub/agarwala/bmtagger

Introduction

The Human Microbiome Project (HMP) is a NIH Roadmap initiative (http://nihroadmap.nih.gov/hmp/) that is part of a larger international effort for studying microbial communities harvested from natural environments found in various human body sites. The sequencing for the project is being done using Illumina and 454 that are both next-generation sequencing technologies producing gigabases of data per day [1]. As of June 16, 2011, searching SRA at NCBI for "Human Microbiome Project" using Entrez gave 3424 experiments of which 3421 had at least one run. Of these 3421 experiments (see details in supplemental table S1), 2291 were sequenced using 454 (16708 runs, 153.8 Gb sequence) and 1130 were sequenced using Illumina (1607 runs, 8270.9 Gb sequence). The sequences generated are "contaminated" by reads that are for the human from whom the sample was collected. A major concern for making the data publicly available is that the human contamination may be sufficient to identify the human subjects. The current policy is to submit all reads to the controlled-access dbGaP Sequence Read Archive (SRA) so that only authorized users can access the full set. For any reads that are determined as human, all their bases are replaced by ambiguity character 'N' in the shadow copy of the submission available publicly in the openly available SRA.

To determine whether a read is human or not, one can align it to the human genome using MegaBlast [2], or one of the various short-read aligners published in the past couple of years. See [5] for a recent survey of published short-read aligners. Determining contamination by alignment is not necessarily the most efficient use of computing resources because we are not interested in finding where and how a read aligns to the human genome, but only in whether it aligns or not. Furthermore, we still need to make a determination for whether a particular alignment is "good enough" and "long enough", and also overcome the issue of missing alignments due to the heuristics used by the short-read aligner. Alignment-free methods for doing sequence comparisons typically use oligomer frequencies or information content (see [7] for a review). To our knowledge, none of the alignment-free methods are appropriate when the two sequences being compared had lengths that differed in orders of magnitude, such as a short read and the human genome in our case. It was also not clear how the published methods would behave when the reads have sequencing errors in them.

We present BMTagger that attempts to first use a heuristic to discriminate between human reads and microbial reads by comparing the 18mers found in the read with those in the human genome. If it fails to make a determination, then it uses an alignment procedure called *srprism* that guarantees to find matches with upto two errors in reads that are at least 32 bp long, if such an alignment exists. For reads of length L longer than 32 bp, *srprism* guarantees to find matches with upto $\lfloor (L/16) \rfloor - 1$ errors. We find that our heuristic is as fast as BWA [3] and much more sensitive than BWA. DeconSeq [4] method published recently modified BWA to make it more sensitive, but is targeting reads lengths longer than the ones produced by Illumina at this time.

The reads can be presented to the tagger as fasta or fastq files, or it can also retrieve reads directly from SRA. SRA design is to divide information for reads into columns, with one column per attribute such as nucleotide sequence and quality. This design leads to easy extensions as the one desired for recording screening status of each read to facilitate appropriate actions when making a public copy.

Materials and Methods

BMTagger screening

BMTagger considers each read individually and decides whether the read can be tagged as human or microbial, or whether the read should be aligned to the human genome in order to make the decision. The determination without alignment is done by *bmfilter* that uses a heuristic for how many and how many consecutive 18mers in the read are also present in the human genome. For reads that are not decided by *bmfilter*, the alignment to the human genome is carried out by *srprism*. The *bmtagger.sh* shell script controls communication between different steps and combines results from each step to produce the final set of results.

Both *bmfilter* and *srprism* require indexes on the human genome to be pre-computed once. Index for *bmfilter* is computed by *bmtool* that generates a bitmap storing information about 18mers present in the human genome. Index for *srprism* is generated by *srprism* command with argument *mkindex*.

We next give some more details for BMTagger, *bmfilter*, and the datasets used for evaluation.

BMTagger algorithm: Steps carried out by BMTagger are as follows:

1. Generate a random string that will be used to name temporary files created in the rest of the process.

2. Generate *bmfilter* command using the input parameters specified. User can specify reads as fasta or fastq files, or specify the SRA run accession as the source of reads. For SRA run accessions, the process uses SRA toolkit [6] to access the reads.

3. Using *bmfilter*, classify reads as foreign or human, and for reads that cannot be classified, generate subsequences of length 32 bp with a distance of 4 bases between the previous and the next, except

the last one if sequence length is not a multiple of 4. Reads shorter than 32 bases are reproduced. Low-complexity subsequences are not generated.

4. Run *srprism* on subsequences for the first mate looking for an alignment with at most one error. If the reads are paired, then remove subsequences (using *extract_fullseq*) for the second mate for reads found as human with first mate and run *srprism* on the rest.

5. Combine outputs from *bmfilter* and *srprism*, and remove temporary files.

***bmfilter* process:** Parameters to *bmfilter* control trimming of reads when ambiguity characters are present. By default, trimming stops when there are no ambiguity characters in the first and last 5 bases. A read (or mate) is considered further by *bmfilter* only if its length after trimming is above a specified length (default of 25 bp). Then, for a sequence of length N, a bitvector of size $(N - 17)$ is generated where the bit at position i is 1 if and only if the 18mer starting at position i is present in the human genome. A similar bitvector is generated for the reverse complement of the sequence. If the number of 1s is less than 20% and the longest run of 1s is less than 10%, then the bitvector is called foreign, if the number of 1s is at least 80% and the longest run of 1s is at least 40%, then the bitvector is called human, else a determination for the bitvector cannot be made. If a determination is made as human for a bitvector, then the read is tagged human regardless of the determination for other bitvectors for the same read. For a read to be tagged foreign, all bitvectors for the read must be called foreign. Note that for paired reads, we can have upto four bitvectors to consider.

Bitvector compression

The 18mer bitvector indicating presence or absence of an 18mer in the human genome contig produces a sparse vector and also put a minimum og 8 Gb memory requirement on the process. We have implemented a couple of compression schemes to reduce the size of the bitvector.

Bitmask takes large space both in file and in memory. For word size of 18 which is found to be practical for Human contamination screening the size is 8 Gigabytes. Therefore a compressed representation of the bitmap was developed as an option. For Human genome build 37 compressed bitmask takes less or about 2.7 Gigabytes in file and same memory is used at screening time. Producing bitmask takes only about 15 min for both contiguous and discontiguous patterns, and takes shorter time for smaller Nmers.

Compressed representation besides header consists of two data parts: fixed size table and sequence of arbitrary size data blocks.

For compression, each word (which in flat bitmask addresses bit with value of 1) is split in prefix and suffix of the predefined sizes. For the Human genome using word size of 18 bases (36 bits) and no ambiguities allowed best compression is achieved with 26 bit prefixes (and 10 bit suffixes).

Prefix is used as an address in the fixed size table. The table contains two columns: number of words sharing this prefix and the address of the data block in the second part of the bitmask. To save space, sizes of these columns can be set in bits and for the discussed case 10 and 34 bits appripriately should be used to achieve best results. Then size of the table is $2^26 * 44$ bits or approximately 360 Megabytes. Smaller values may end up with everflow error, larger will unnecessarily increase primary table size.

Number of words sharing same prefix is used to determine size of the secondary block. For some prefixes number of suffixes available for the genome may be very short, and in this case list of suffixes may take just few bytes. Extreme case is empty list which takes no space. For other prefixes list of suffixes may take more space then bitmap; in this case data block contains bitmap. Since suffix size is predefined, so is the maximal number of suffixes when list takes less space then bitmap. In the discussed case list of $102(2^10/10)$ 10-bit suffixes takes as much space (in bytes) as bitmask addressed with 10 bits (1 kB), so whenever we have shorter list we win in space compared to a bitmap block.

In these terms flat bitmask representation is a special case of the compressed ones with suffix length of 0, word count length of 1 and data offset length of 0.

To check if a word is present in the reference genome following algorithm is used: 1st, prefix is computed; 2nd, table entry (count, offset) is retrieved; 3rd - if length of the list is 0, then algorithm finishes knowing that the word is absent; else 4th - for very short list it is consecutively scanned till match of the list item to the suffix of the query word or to the end; for longer lists binary search is used; or if the value of number of suffixes in the table is so large that the datablock contains bitmask - the bit value which corresponds to the suffix value is checked.

Somewhat higher level of compression may be achieved if it is taken into account that some prefixes may share same list of suffixes. In this case same data block can be addressed by two or more table entries. Detecting such cases noticeably increases CPU time while compressing bitmask.

0.1 Evaluation datasets

0.2 Evaluation method

run bmtagger

bwa command line

48 mer with at most 2 errors

Results

Choice of Nmer and pattern for heuristic

Accuracy assessment

The number of foreign and human reads in the benchmark sets is given in Table 1. Results on the benchmark sets and five SRA accessions are presented in Tables 2 and 1. Our experiments show that discriminating between human and microbial reads can be done efficiently without resorting to time-consuming alignment of all reads to the human genome, as shown by the classification done by *bmfilter* in Table 2 and low error rate (defined as number of reads tagged incorrectly as compared to the total number of reads) shown in Table 1.

The human set has 3928 reads (1977 on forward and 1951 on reverse strand) where only the read or its reverse complement is tagged as human and the other one is not. This inconsistency is because, with the stride of 4 and length not a multiple of 4, the 32mers generated for alignment by *srprism* are different for the read and its reverse complement.

NOTE: RICHA ADD RESULTS FOR ADDITIONAL SIMULATED SETS

First compression scheme reduces the memory footprint of the process from the current 8.1 Gb to 2.6 Gb. This scheme reduces running time for runs when the running time is dominated by the time taken for loading the index for *bmfilter* (such as for SRR040576), but typically takes 50% longer for runs in Table 2.

Running time

compressed better for really small sets

We did not investigate DeconSeq extensively, but submitting SRR14.. gave us the message that “The running step will take about 10 minutes for each 100 MB of submitted data independent of the number of databases selected (assuming the cluster has enough free resources at the time of computation).”

Discussion

HMP project has several centers producing sequences. A consistent treatment of all sequences for contamination screening provides opportunities to both improve screening method, and also adequaltely compensate for any shortcomings in the screening method. As sequences are submittted to controlled SRA, it was natural for NCBI to undertake development of a method for screening the human reads and to provide microbial reads openly to the public. The pipeline for doing these tasks utilizing BMTagger has been in production at NCBI since July 2010.

NOTE: RICHA ADD TEXT AND RESULTS FOR DISCONTIGUOUS PATTERNS AND SHORTER KMERS

We find that most of the reads tagged incorrectly are low-complexity reads. We plan to investigate the effect of using the human genome that is masked for low-complexity. We are also working on incorporating improvements in *srprism* that for reads of length L where $L > 32bp$, now guarantees to find alignments with up to $\lfloor (L/16) \rfloor - 1$ errors.

Acknowledgments

We thank our colleagues Kurt Rodarmer, Andriy Klymenko, Michael Kimelman, and Eugene Yaschenko for incorporating BMTagger in the NCBI pipeline for screening reads for HMP and Steve Sherry for communicating the needs of the HMP that led to the development of BMTagger. We also thank Mitreva Makedonka (Washinton University, St. Louis) for providing the sets used in the initial development of BMTagger and for useful discussions about the alignment based methods they are using. Sudeep Mehrotra (McGill University, Montral) provided helpful comments for improving documentation and also sent the reference for DeconSeq. This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References

1. Metzker,M.L. (2010) Sequencing technologies - the next generation, *Nature Reviews Genetics*, **11(1)**, 31–46.

2. Zhang, Z., Schwartz, S., Wagner, L., and Miller, W. (2000) A greedy algorithm for aligning DNA sequences, *J. Comp. Biol.*, **7**, 203–214.

3. Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform, *Bioinformatics*, **25(14)**, 1754–1760.

4. Schmieder, R. and Edwards, R. (2011) Fast identification and removal of sequence contamination from genomic and metagenomic datasets, *PLoS One*, **6(3)**, e17288.

5. Li, H. and Homer, N. (2010) A survey of sequence alignment algorithms for next-generation sequencing, *Briefings in Bioinformatics*, doi:10.1093/bib/bbq015.

6. SRA Toolkit: http://www.ncbi.nlm.nih.gov/Traces/sra/
sra.cgi?cmd=show&f=software&m=software&s=software

7. Vinga, S., and Almeida, J. (2003) Alignment-free sequence comparison – a review, *Bioinformatics*, **19(4)**, 513–523.

Figure Legends

Tables

Table 1. BMTagger error rate

Dataset	Foreign	Human	F as H	H as F	Error
metagenomic	508149	50815	0	448	0.080
confirmed	40000000	0	6873	na	0.017
simulated	40000000	0	12785	na	0.031
q2trimmed	39991384	4000000	4006	64935	0.157
human	0	10000000	na	203030	2.030

Number of foreign and human reads in benchmark sets and the number of reads incorrectly classified. Columns labeled ”F as H” and ”H as F” given the number of foreign reads classified as human and the number of human reads classified as foreign, respectively.

Table 2. bmfilter classification

	Classification by *bmfilter*			
Dataset	Foreign	Human	Unknown	Running time (min)
SRR006003	12670	1281162	342827	24 (9)
SRR046896	11258	2	42	0.7 (0.3)
SRR051944	998981	52164	97182	7 (6.4)
SRR059818	38183094	98188	11616610	185 (114)
SRR059480	23364262	5978338	6378194	108 (64)
metagenomic	508269	44969	5726	3.5 (2.2)
q2trimmed	34880460	3381760	5729164	65 (47)
confirmed	34838129	3865	5158006	54 (44)
simulated	36025143	8330	3966527	65 (44)
human	149990	7008934	2841076	32 (9)

Number of reads classified as foreign or human, and the number of reads that cannot be classified by *bmfilter* is shown. Total time taken by BMTagger, with the time taken by *bmfilter* in brackets is also shown.